Syllabus

Spring 2022 -- COMP 380 -- Practical Problem Solving

Goals

This course pursues multiple distinct goals, which are directionally aligned.

In short: after this class you will:

- Become a better Java programmer.
- Start believing in Probabilities and Statistics :-)
- And gain a better understanding of, and experience with, Machine Learning.

As mentioned in the Pitch, the problem domain we will work with is the card game of Blackjack.

A more in-depth look at what we will cover:

- Improve your Java coding and related SWE skills:
    - Practice and improve your skills in Object-Oriented design;
    - Get exposure to, and understanding of, more advanced features of Java programming, such as *JMM* (Java Memory Model), *concurrency* and *thread-safety*, and *futures-based design patterns*;
    - Become familiar with more advanced Java libraries;
    - Learn to use Github.

- Use your Java OO design skills to model a somewhat complex universe of the card game of Blackjack: which entities exist there, how they interact depending on the situation, etc.

- Experiment with algorithms for playing Blackjack, design and develop an experimentation system, where individual algorithms could be quickly compared by their quality. You could add some heuristics, based on your intuition, and instantaneously see whether they work!

- Work with probability calculations to develop a Mathematically best algorithm for the game. A "brute force" approach here can quickly become messy and also slow, so we will look at how such algorithms could be expressed in a clear, human-readable way, which also could execute much faster.

- As a final goal, we will approach this problem from the Machine Learning angle: instead of implementing heuristics or probability calculations, we will throw raw data at an ML algorithm and build models. We would expect this model to perform slightly below our "Mathematically perfect" one.

There is no specific goal to make you experts in Blackjack – but you might become a more skilled player and you will understand that Casinos win :-) To make sure, we will all need to agree on some version of Blackjack, to make comps meaningful, and you will learn the rules of the game.

Implementing GUI for the game is not a requirement – but you can do it for fun.

How

The course is hands-on and you will be involved in writing code from day one.

The instructor will help you understand some basic principles or offer ideas – but it will be your job to implement it.

Your progress will be evaluated in several ways:

- **Code Quality**: at some "checkpoints", the instructor and TAs will read your code, comment on it, and assign a score. We will look at:
  - **correctness**
  - **simplicity** – given that the domain is complex, not adding unnecessary complexity is important
  - **readability** – appropriate comments, javadocs, etc.
  - **testing**: is your code sufficiently tested?
- **Automated evaluation**: when you present your implementation of a certain algorithm, we could:
  - Determine its correctness by running it alongside the "Golden" version.
  - Runtime: is it unusually slow? Why?
- **Presentations/questions** (see below)

Alexei likes the class to be very interactive; he likes to be interrupted, esp. If he's talking nonsense or is otherwise unclear. Good questions or suggestions could bring you bonus points.

Experimental Nature

This class, in this form, is taught for the first time, so apologies for any rough edges or on-the-fly changes. We do need feedback from the class if something doesn't work.

For starters, we haven't quite decided whether partitioning of students into pairs or triplets would work better than individually – we appreciate feedback from you.

I would like to give students opportunities to present their progress publicly, and earn points for this, but without grouping this may be hard to run.

Grades

The final letter grades will be assigned by applying a curve to numerical "point scores" earned by students.

General Information

Class's schedule **MWF 11 AM -11:50 AM, Duncan Hall 1075**

Instructor Dr. Alexei Stolboushkin, aps@rice.edu
office hours: TEMPORARY: By appointment (send email)

TAs **Pratik Satish,** Pratik.Satish@rice.edu
Durga Parulekar, Durga.K.Parulekar@rice.edu

	Aditya Dave, aditya.dave@rice.edu Office Hours: Aditya Dave: TEMPORARY: By appointment (send email) Durga Parulekar: TEMPORARY: By appointment (send email)
Overall course objective and expected learning outcomes	See the Goals section above.
Grade policies	See the Grades section above.
Absence policies	**Liberal.**
List of required texts	Not absolutely required, but I highly recommend that you buy *"Effective Java"* by Josh Bloch. On Amazon, it cost $42.50 for the newest (2017) edition.
Special materials required for this class	A personal laptop is required. Having a Macbook (Air or Pro or above) is preferred, but Windows or Chromebook laptops may work. Since the laptops are to be used for development, the cheapest Chromebook models targeted at middle-schoolers might not suffice: you need some RAM and processing power (CPU) to comfortably handle the development tasks.
Required examination and papers	See the How section above.
Statement of expectations regarding coursework	As described above, it is a challenging class, where your success builds on the success of your team and, in addition, requires your active participation during the class time through presentations, discussions, and other interactions, as well as through your self-evaluation and giving and receiving peer feedback. All students commit to the Rice Honor Code.
Special needs	Rice is committed to providing the best educational environment to all of our students and we are particularly cognizant of the students who may have special needs due to a disability. In this class, and in general, we will do our best to provide all reasonable accommodations to students who are disabled or have other special needs. Please contact Alexei to request such specific special accommodations and rest assured that your requests will be handled discreetly and in a timely manner. Rice's Disability Resource Center (DRC) provides more information and resources.

The information contained in this Syllabus is subject to change, based on circumstances and as deemed appropriate, with reasonable advance notice. In particular, the number of expected presentations will depend on the number of students and teams.